We claim:

1. An apparatus for performing virtual identification (VID) to physical identification (PID) translation for data elements to be accessed within local memory of a processing element whereby a direct memory access controller can access PE memories according to their VIDs, the apparatus comprising:

a DMA controller; and

a memory maintained in the DMA controller for storing a processing element VID-to-PID table mapping processing element VIDs to processing element PIDs.

2. The apparatus of claim 1 wherein said memory is maintained in a core transfer unit of the DMA controller.

3. The apparatus of claim 2 wherein the core transfer unit (CTU) further comprises an address generation unit (AGU) which receives a CTU transfer instruction which specifies a starting address which is used by the AGU to generate an initial VID.

4. The apparatus of claim 3 wherein the initial VID controls the selection of one of the elements of the VID-to-PID lookup table through a multiplexer.

5. The apparatus of claim 4 further comprising a DMA bus for providing the selected PID as a first component of a PE address.

6. The apparatus of claim 5 wherein the AGU further operates to generate a PE memory offset which is sent as a second component of a PE address on the DMA bus.

7. The apparatus of claim 6 further comprising a local memory interface unit (LMIU) which is used to compare the PID sent on the DMA bus to a stored PID for any DMA access, if a match is detected then the LMIU accepts the access.

8. The apparatus of claim 3 wherein successive VIDs are generated in recursive fashion by the AGU.

9. An apparatus for performing VID-to-PID translation for data elements to be accessed within local memory of a processing element whereby a direct memory access controller can access PE memories according to their VIDs, the apparatus comprising:

a DMA controller; and

translation logic which performs a subset of all VID-to-PID mappings.

10. The apparatus of claim 9 wherein said translation logic is located in a core transfer unit (CTU) of the DMA controller.

11. The apparatus of claim 10 wherein the CTU further comprises an AGU which receives a CTU transfer instruction which specifies a starting address which is used by the AGU to generate an initial VID.

12. The apparatus of claim 11 wherein the initial VID is mapped to a PID by the translation logic.

13. The apparatus of claim 12 further comprising a DMA bus for providing the translated PID as a first component of a PE address.

14. The apparatus of claim 3 wherein successive VIDs are generated in recursive fashion by the AGU, and further comprising:

a local memory interface unit for each processing element (PE) storing a VID for each PE.

15. The apparatus of claim 14 wherein a VID available to a particular LMIU or a DMA bus is compared with the stored VID in the LMIU and where a match occurs the LMIU accepts the access.

16. A method of providing complex data access patterns for accessing the memory elements of multiple processing elements (PEs) in a DMA controller, the method comprising the steps of:

defining a PE memory address comprising three address components, a PE virtual ID (VID), a base value and an index value;

assigning each address component to one of a number of nested logical loops, an inner, a middle and an outer logical loop;

updating address components by selecting new values for the address components;

indicating each logical loop's exit condition; and

initializing address components upon reaching a loop exit condition.

17. The method of claim 16 wherein a first component is assigned to the inner logical loop; looping occurs until a number of index updates is exhausted; and the loop exit condition is detected at which point the index is reinitialized.

18. The method of claim 17 wherein subsequent to reinitialization of the index, a second component is assigned to the middle logical loop.

19. The method of claim 18 wherein looping continues until a transfer count is exhausted.

20. The method of claim 16 further comprising the step of rearranging the order of assignment of address components to the logical loops.

21. The method of claim 16 further comprising the step of varying the way of updating the address components.

22. The method of claim 16 further comprising the step of varying the loop exit conditions.

23. A system for providing complex data access patterns for accessing the memory elements of multiple processing elements (PEs) in a DMA controller, the system comprising:

means for defining a PE memory address comprising three address components, a PE virtual ID (VID), a base value and an index value;

means for assigning each address component to one of a number of nested logical loops, an inner, a middle and an outer logical loop;

means for updating address components by selecting new values for the address components;

means for indicating each logical loop's exit condition; and

means for initializing address components upon reaching a loop exit condition.

24. The system of claim 23 wherein a first component is assigned to the inner logical loop, and the means for looping continues to loop until a number of index updates is exhausted and the loop exit condition is detected at which point the index is reinitialized.

25. The system of claim 23 wherein said means for assigning comprises a programmer selected ordering of loop address components by parameters in a transfer instruction.

26. The system of claim 24 wherein subsequent to reinitialization of the index, a second component is assigned to the middle logical loop.

27. The system of claim 26 further comprising a transfer count register and wherein looping continues until a transfer count is exhausted.

28. The system of claim 23 further comprising means for rearranging the order of assignment of address components to the logical loops.

29. The system of claim 23 further comprising means for varying the way of updating the address components.

30. The system of claim 23 wherein said means for updating comprises means for selecting an entry from a table of update values.

31. The system of claim 23 wherein said means for updating comprises means for adding a specified value to the old address value and using this computed value as the new value.

32. The system of claim 23 further comprising means for varying the loop exit conditions.

33. The apparatus of claim 23 further comprising:

means for determining that a requested number of accesses have been made and exiting all loops immediately leaving all address and loop control variables in their current state thereby allowing loop entry and transfers to be restarted so that addressing continues as it would if the transfer count had not been exhausted.

34. The apparatus of claim 23 wherein the means for updating operates to update addresses by using a constant increment value.

35. The apparatus of claim 23 wherein the means for updating operates to update utilizing an increment value extracted from a table stored in a memory.

36. The apparatus of claim 23 wherein the means for updating employs a selection mechanism based upon a bit vector.

37. The apparatus of claim 23 wherein a loop parameter specifies the assignment of address components to loops.